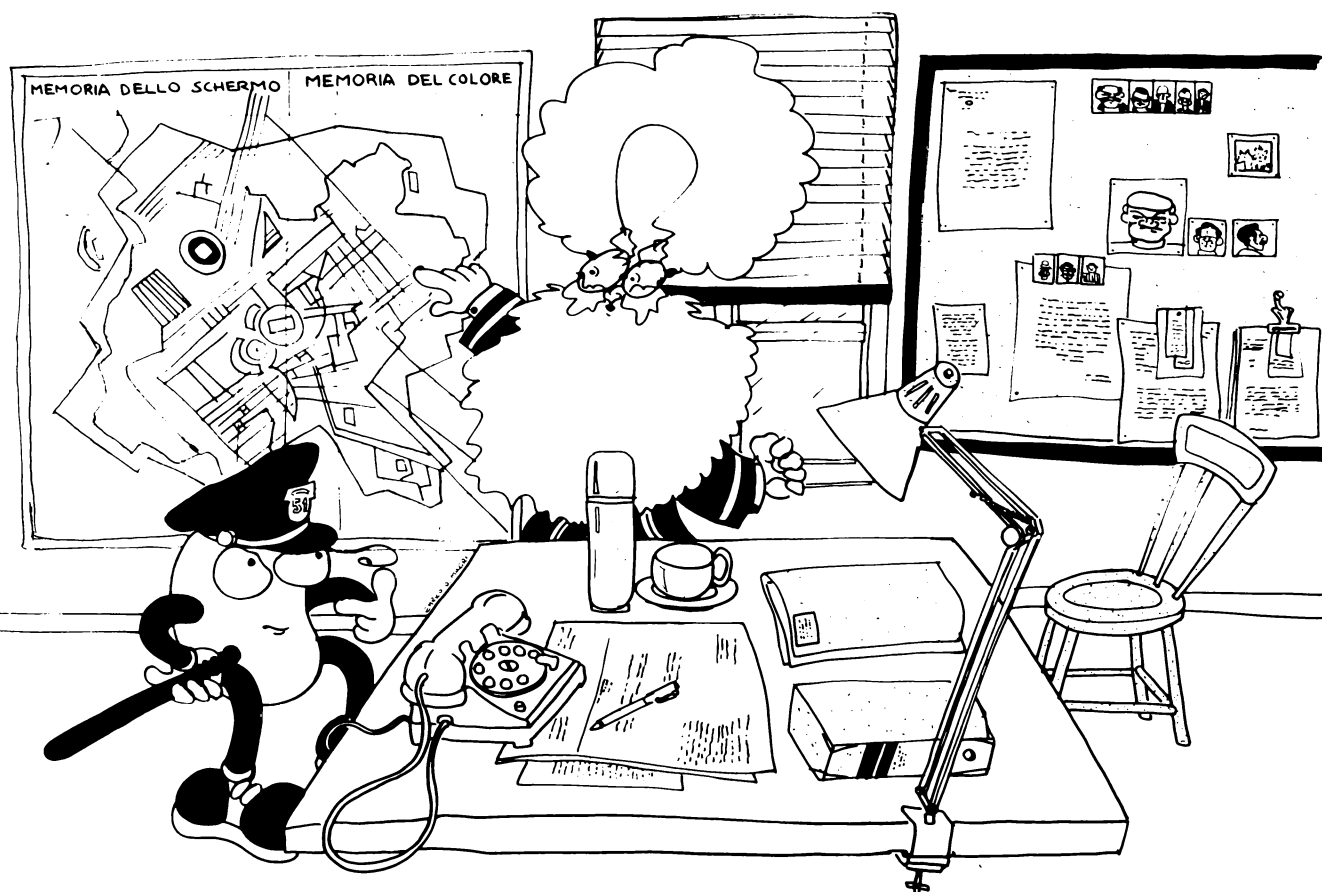
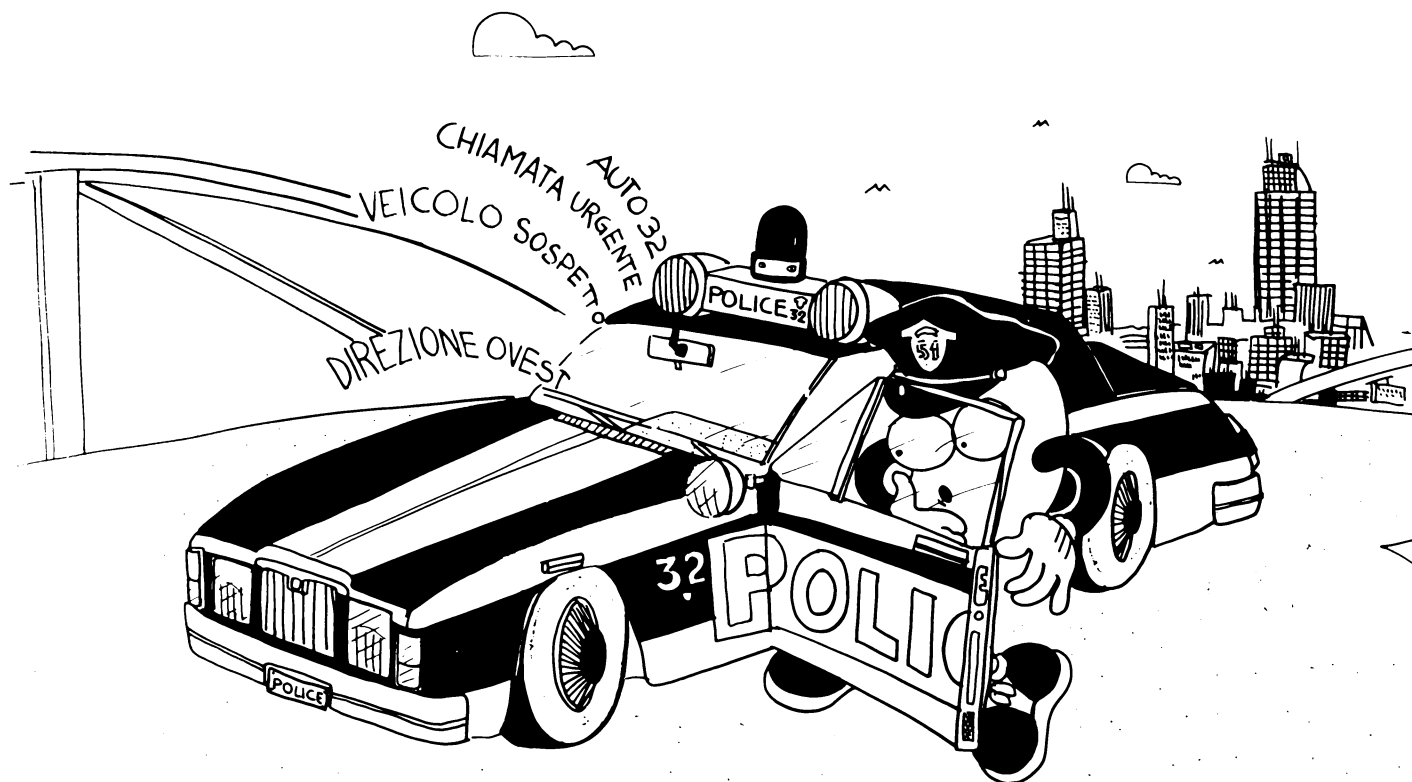
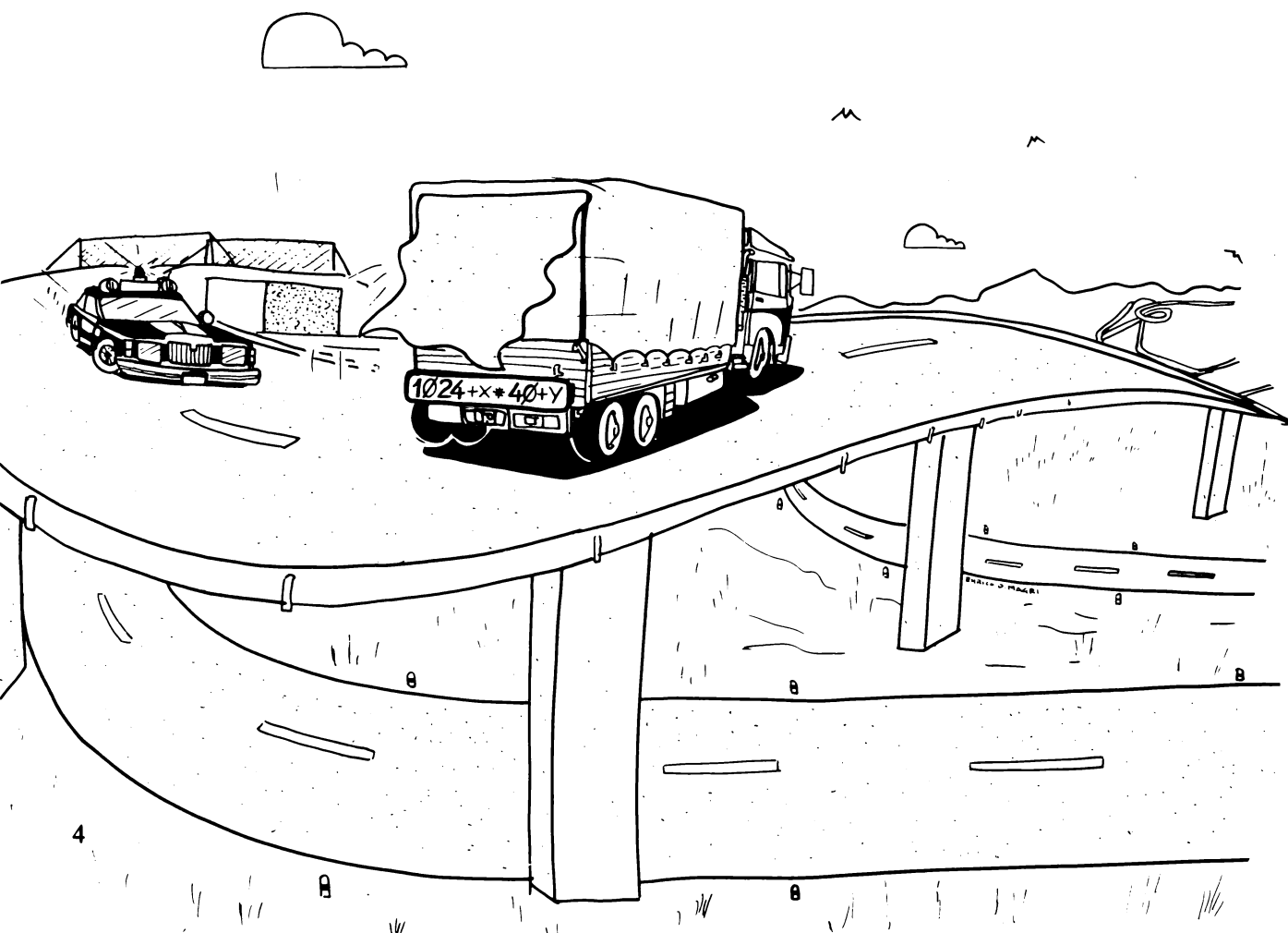
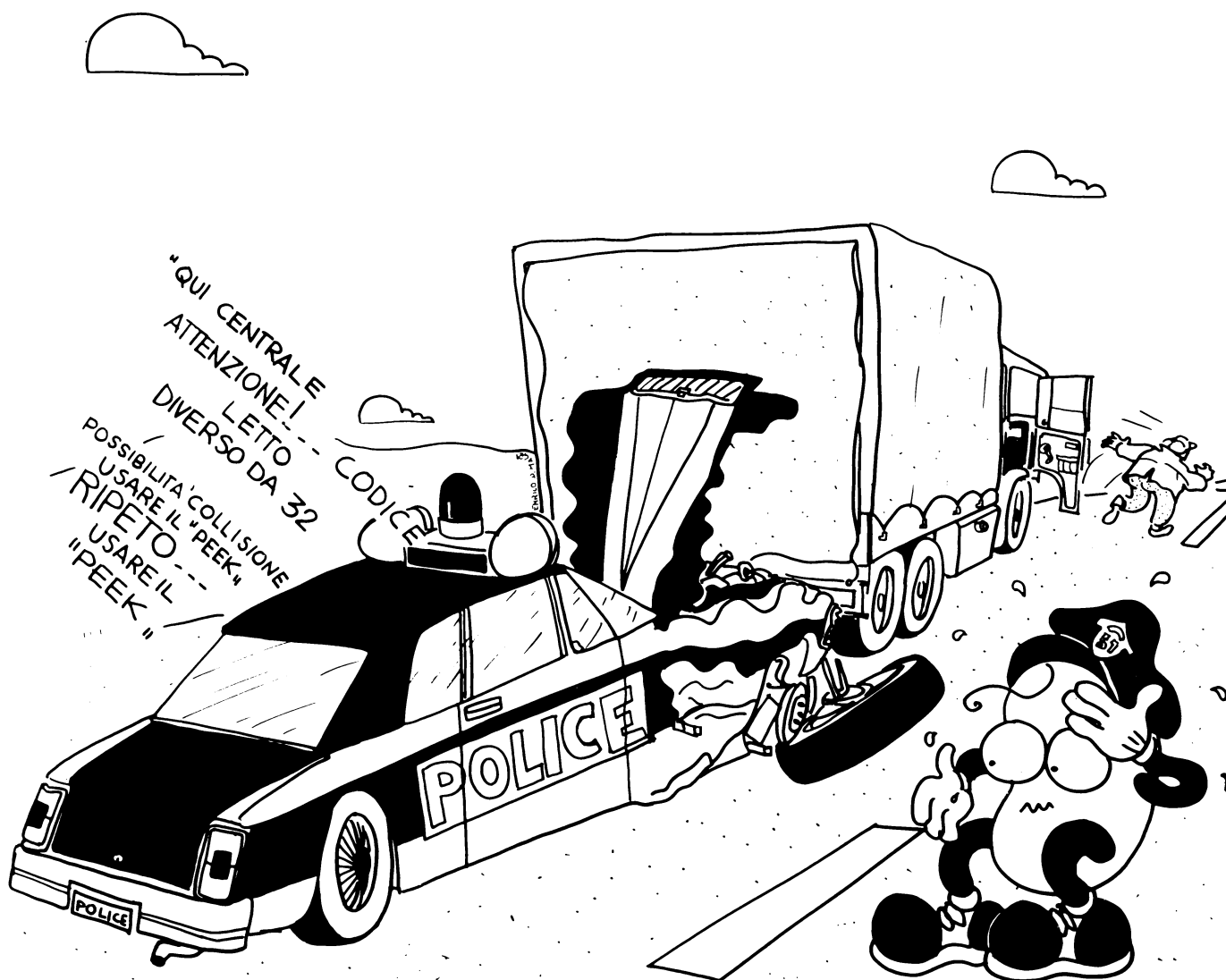
CONOSCERE IL COMPUTER DIRETTAMENTE DAL COMPUTER

per Commodore Vic20 e 64

Beatrice d'Este

Nel COMPUTER esistono due aree di memoria, chiamate MEMORIA DELLO SCHERMO e MEMORIA DEL COLORE.
LA MEMORIA DELLO SCHERMO contiene i codici dello schermo (o di visualizzazione) dei caratteri che sono visualizzati sul video.
Ad ogni posizione del video, riga per riga, è associata una locazione di memoria che conterrà il codice del carattere presente nella corrispondente posizione.
Quindi la prima locazione conterrà il codice del carattere presente nell'angolo in alto a sinistra del video, mentre l'ultima quello nell'angolo in basso a destra.

Naturalmente se in una posizione non è presente nessun carattere, nella corrispondente locazione sarà presente il codice dello spazio, cioè 32.
LA MEMORIA DEL COLORE contiene i codici dei colori dei caratteri in ogni posizione dello schermo.
Per visualizzare un carattere con il POKE ne memorizzerai il codice nella locazione della memoria video, corrispondente alla posizione in cui dovrà apparire.

Se vuoi indirizzare le posizioni dello schermo tramite le coordinate X e Y puoi calcolare la locazione corrispondente con la formula:
(per il CBM 64) 1Ø24+X*4Ø+Y.
Dove X è variabile tra Ø e 24 e Y tra Ø e 39, visto che sul video ci sono rispettivamente 25 righe e 4Ø colonne.
(per il VIC 20) 768Ø+X*22+Y
Dove X è variabile tra Ø e 22 e Y TRA Ø E 21, dato che ci sono sul video rispettivamente 23 righe e 22 colonne.
Sul video puoi creare il movimento di un carattere semplicemente memorizzandone il codice della nuova posizione del video e cancellando quello nella posizione precedente con il valore 32 (codice dello spazio).

Quindi, con l'istruzione PEEK potrai leggere il contenuto di una locazione del video o del colore (lez. 25), così da sapere quale carattere o quale colore è presente in una certa posizione dello schermo.
Questo è molto utile per stabilire se due caratteri si scontrano.
Quindi, durante il movimento di un carattere, prima di spostarlo alla posizione successiva, con il PEEK è possibile leggerne il codice.
Se il codice letto è diverso da 32 significa che il carattere è in collisione con un altro.

Listato dell'esercizio: ESEMPIO PRATICO (CBM 64)

```
10 poke53280,6:poke53281,6
15 printchr$(142)chr$(147)
20 v=1024:c=55296
30 x=5:fory=10to30
40 gosub500:pokev+p,81:pokec+p,7
50 nexty
60 y=30:forx=5to20
70 gosub500:pokev+p,81:pokec+p,7
80 nextx
90 x=20:fory=30to10step-1
100 gosub500:pokev+p,81:pokec+p,7
110 nexty
120 y=10:forx=20to5step-1
130 gosub500:pokev+p,81:pokec+p,7
140 nextx
150 x=8:fory=25to15step-1
160 gosub500:pokev+p,81:pokec+p,1
170 nexty
180 y=15:forx=8to17
190 gosub500:pokev+p,81:pokec+p,1
200 nextx
210 x=17:fory=15to25
220 gosub500:pokev+p,81:pokec+p,1
230 nexty
240 y=25:forx=17to8step-1
250 gosub500:pokev+p,81:pokec+p,1
260 nextx:end
500 p=x*40+y:return
```

Listato dell'esercizio: ESEMPIO PRATICO (VIC 20)

```
10 poke36879,110
15 printchr$(142)chr$(147)
20 v=7680:c=38400
30 x=4:fory=2to20
40 gosub500:pokev+p,81:pokec+p,7
50 nexty
60 y=20:forx=4to18
70 gosub500:pokev+p,81:pokec+p,7
80 nextx
90 x=18:fory=20to2step-1
100 gosub500:pokev+p,81:pokec+p,7
110 nexty
120 y=2:forx=18to4step-1
130 gosub500:pokev+p,81:pokec+p,7
140 nextx
150 x=6:fory=16to6step-1
160 gosub500:pokev+p,81:pokec+p,1
170 nexty
180 y=6:forx=6to16
190 gosub500:pokev+p,81:pokec+p,1
200 nextx
210 x=16:fory=6to16
220 gosub500:pokev+p,81:pokec+p,1
230 nexty
```

```
240 y=16:forx=16to6step-1
250 gosub500:pokev+p,81:pokec+p,1
260 nextx:end
500 p=x*22+y:return
```

Listato dell'esercizio: COMPLETA IL LISTATO (CBM 64)

```
10 poke53280,6:poke53281,3:[*]=54272
20 print[*](142)chr$(147)
30 for[*]=40to999
40 ifint([*](0)*50)<46then60
50 poke1024+x,[*]:poke55296+x,6
60 nextx:[*]=1064:a=ti
70 [*]a$:pokek,88:pokec+k,1:pk=k
80 ifa$="x"then k=[*]
90 ifa$="z"then k=k-1
100 if[*]=";"then k=k-40
110 ifa$="/"then k=k+40
120 ifpk<>kthenpokec+pk,3
130 [*](k<1064)+(k>2023)thenk=1064
140 ifpeek(k)=86then[*]=s+1:pokec+k,3
150 printchr$(19)tab(15)"punti:";s
160 ifti-[*]<1800then70
```

Listato dell'esercizio: COMPLETA IL LISTATO (VIC 20)

```
10 poke36879,62:[*]=30720
20 print[*](142)chr$(147)
30 for[*]=22to505
40 ifint([*](0)*50)<46then60
50 poke7680+x,[*]:poke38400+x,6
60 nextx:[*]=7702:a=ti
70 [*]a$:pokek,88:pokec+k,1:pk=k
80 ifa$="x"then k=[*]
90 ifa$="z"then k=k-1
100 if[*]=";"then k=k-22
110 ifa$="/"then k=k+22
120 ifpk<>kthenpokec+pk,3
130 [*](k<7702)+(k>8185)thenk=7702
140 ifpeek(k)=86then[*]=s+1:pokec+k,3
150 printchr$(19)tab(6)"punti:";s
160 ifti-[*]<1800then70
```

PROGRAMMIAMO INSIEME (CBM 64)

```
10 poke 53280,0:poke 53281,0
20 dim d(100)
30 printchr$(142)
40 print"[illegible]numero stelle"
50 input"[illegible](1-100)";ns
60 if (ns<1)+(ns>100) then 40
70 print"[illegible]"
80 for k=1 to ns
90 ps=int(rnd(0)*1000)
100 if peek(1024+ps)<>32 then 90
```

```
110 d(K)=ps
120 cs=int(rnd(0)*15+1)
130 ts=int(rnd(0)*3+1)
140 if ts=1 then cr=81
150 if ts=2 then cr=42
160 if ts=3 then cr=46
170 poKe 1024+ps,cr
180 poKe 55296+ps,cs
190 next K
200 for K=1 to ns
210 poKe 1024+d(K),32:poKe 55236+d(K),0
220 for t=1 to 80:next t
230 next K
```

PROGRAMMIAMO INSIEME (VIC 20)

```
10 poKe 36879,8
20 dim d(50)
30 printchr$(142)
40 print"[illegible]numero stelle"
50 input"[illegible](1-50)";ns
60 if (ns<1)+(ns>50) then 40
70 print"[illegible]"
80 for K=1 to ns
90 ps=int(rnd(0)*506)
100 if peeK(7680+ps)<>32 then 90
110 d(K)=ps
120 cs=int(rnd(0)*7+1)
130 ts=int(rnd(0)*3+1)
140 if ts=1 then cr=81
150 if ts=2 then cr=42
160 if ts=3 then cr=46
170 poKe 7680+ps,cr
180 poKe 38400+ps,cs
190 next K
200 for K=1 to ns
210 poKe 7680+d(K),32:poKe 38400+d(K),0
220 for t=1 to 80:next t
230 next K
```

Soluzione dell'esercizio: COMPLETA IL LISTATO (lez. 25)

```
10 forx=1to6:readw$(x):nextx
20 data"for x=1 to 10"
30 data"goto 110"
40 data"input n"
50 data"s=s+1"
60 data"next j"
70 data"if a<>2 then 100"
80 s=ti:forx=1to6
90 r=int(rnd(0)*6+1)
100 if v(r)=1 then 90"
110 v(r)=1
120 print:printtab(2)w$(r):print
130 inputs$:ifs$=w$(r)then150
140 goto120
150 nextx:f=ti:print
160 print"secondi:"int((f-s)/60)
```